

## **REMARKS**

Claims 1, 3, and 5 stand rejected under 35 U.S.C. 102(e) as being anticipated by Guedalia et al. (U.S. Patent No. 6,535,878). Claim 1 is amended to incorporate the subject matter of claim 2, claim 3 is amended to incorporate the subject matter of claim 4, and claim 5 is amended to incorporate the subject matter of claim 6. Accordingly, the rejection is moot.

Claims 2, 4 and 6 stand rejected under 35 U.S.C. 103(a) as being obvious over Guedalia, in view of Jagannathan et al. (U.S. Patent No. 5, 692,193). Applicant respectfully traverses the rejection because the cited references fail to disclose or suggest a thread-creation-request holding unit and a second thread execution unit, as recited in amended claims 1 and 5. With respect to amended claim 3, Applicant traverses the rejection because the cited references fail to disclose or suggest the thread-creation-request holding unit and a second thread executing unit that perform the step of processing a request to create a thread that is not executed because an operation to process the request failed, by reusing an arrangement provided for supporting and managing another thread, and without creating a new thread.

The Examiner cites Guedalia. Guedalia teaches a method for limiting the number of concurrent threads implemented in servers. A maximum number of threads is pre-set, and multiple threads are created as requests to create them are received. Thread initiation is determined by client requests. The threads form a thread pool, and can be in either of two

states: i) a wait state, in which the threads are idle and wait for a request to process; or ii) a busy state, in which the threads are processing a request. When the maximum limit of allowable concurrent requests/threads is reached and the threads are all busy, subsequent requests are queued pending availability of a free thread. (Col. 4, lns. 28-41).

The Examiner asserts that Guedalia teaches that when execution of another thread by a first thread execution unit is completed, a second thread execution unit executes a process corresponding to a request held by a thread-creation-request holding unit, without creating a new thread in response to a request held by the thread-creation-request holding unit. Applicant traverses this statement of the Examiner. As discussed above, Guedalia teaches that subsequent requests are merely queued when an overflow condition occurs. Moreover, threads are created as requests are received. Therefore, when a request reaches the end of the queued, thread creation occurs, unlike the present invention. That is, the present invention stores a request from a first unit to a second unit, and then processes the request using the second unit. Guedalia merely processes a request using a first unit.

With respect to claim 3, Applicant traverses the rejection for the reasons recited above, namely, because a step of executing processing of a request is performed to create a thread by reusing an arrangement that supports and manages another thread, without recreating a thread, is not disclosed or suggested by the cited references.

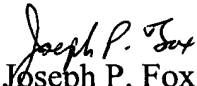
For these reasons, withdrawal of the §103 rejection of amended claims 1, 3, and 5 is respectfully requested.

For all of the foregoing reasons, Applicant submits that this Application is in condition for allowance, which is respectfully requested. The Examiner is invited to contact the undersigned attorney if an interview would expedite prosecution.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By

  
Joseph P. Fox  
Registration No. 41,760

May 28, 2004

300 South Wacker Drive - Suite 2500  
Chicago, Illinois 60606  
Telephone: (312) 360-0080  
Facsimile: (312) 360-9315  
Customer Number 24978

P:\DOCS\0828\65156\545424.DOC